

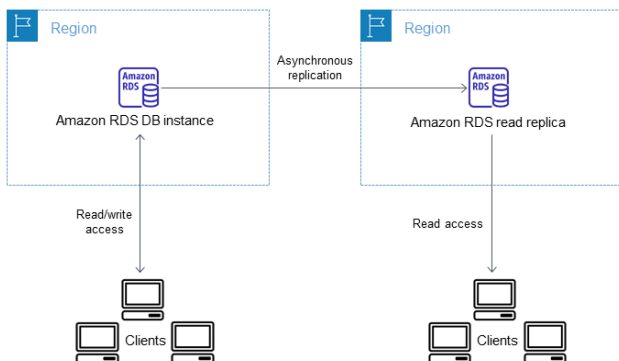
## 5. Database

### Amazon Databases Relacionais

Há dois conceitos primordiais de se entender sobre databases relacionais dentro da AWS, que são Multi-AZ e Read Replica.

**Multi-AZ:** neste sistema, o webserver estará apontando para o database em uma AZ, caso aquela AZ esteja indisponível, o webserver apontará para o DNS da réplica automaticamente.

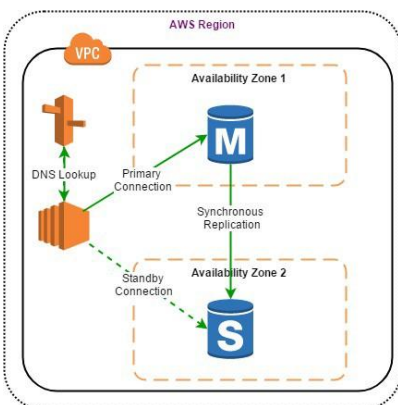
**Read Replica:** neste sistema o webserver estará apontando para um database, porém caso ocorra uma indisponibilidade o webserver não aponta automaticamente para a réplica, é necessário fazer um apontamento manual. A vantagem deste sistema é que é possível ter Auto Scaling Group e alguns servidores apontarem para a réplica, fazendo um balanceamento de tráfego e distribuindo a quantidade de acessos nos databases.



#### Multi-AZ vs Read Replica

O Amazon RDS Multi-AZ e as réplicas de leitura mantêm uma cópia do banco de dados, mas são de natureza diferente. Use implantações Multi-AZ para alta disponibilidade/failover e réplicas de leitura para escalabilidade de leitura.

#### Multi-AZ



As implantações do Amazon RDS Multi-AZ fornecem disponibilidade aprimorada para instâncias de banco de dados em uma única região da AWS. Com Multi-AZ, seus dados são replicados de forma síncrona para um modo de espera em um AZ diferente.

No caso de uma falha de infraestrutura, o Amazon RDS executa um failover automático para o modo de espera, minimizando a interrupção de seus aplicativos sem intervenção administrativa.

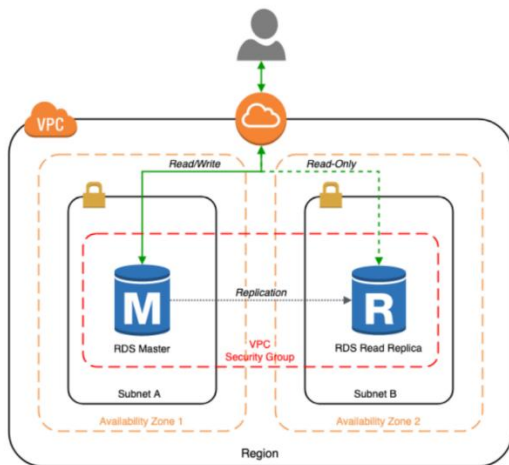
#### Benefits of Multi-AZ deployment:

- A replicação para uma réplica em espera é síncrona, o que é altamente durável.

- O endpoint da instância do banco de dados permanece o mesmo após um failover, o aplicativo pode retomar as operações do banco de dados sem intervenção manual.
- Se ocorrer uma falha, o impacto da disponibilidade é limitado ao tempo que o failover automático leva para ser concluído. Isso ajuda a alcançar maior disponibilidade.
- Reduz o impacto da manutenção. O RDS executa a manutenção no modo de espera primeiro, promove o modo de espera para mestre primário e, em seguida, realiza a manutenção no antigo mestre que agora é uma réplica em espera.
- Para evitar qualquer impacto negativo do processo de backup no desempenho, o Amazon RDS cria um backup da réplica em espera.
- Quando um problema é detectado na instância primária, ele fará failover automaticamente para o standby nas seguintes condições: 1) A instância de banco de dados primária falha. 2) Uma interrupção da zona de disponibilidade. 3) O tipo de servidor da instância de banco de dados é alterado. 4) O sistema operacional da instância do banco de dados está passando por um patch de software. 5) O failover manual da instância do banco de dados foi iniciado usando a reinicialização com failover.

Cross-region Multi-AZ is not currently supported yet.

### Read Replica



As réplicas de leitura do Amazon RDS permitem que você crie uma ou mais cópias somente leitura de sua instância de banco de dados na mesma região da AWS ou em uma região da AWS diferente para aumentar a escalabilidade.

As atualizações feitas no banco de dados de origem são copiadas de forma assíncrona para as réplicas de leitura. As gravações podem acontecer apenas no banco de dados principal e as leituras podem acontecer no banco de dados de réplica de leitura.

Ao criar uma réplica de leitura, você primeiro especifica uma instância de banco de dados existente como a origem. Em seguida, o Amazon RDS tira um instantâneo da instância de origem e cria uma instância somente leitura a partir do instantâneo. O banco de dados de origem deve ter backups automáticos ativados para configurar a réplica de leitura.

### Benefits of Read Replicas

- Réplicas de leitura ajudam a diminuir a carga no banco de dados primário atendendo ao tráfego somente leitura.
- Você pode criar réplicas de leitura em AZ, Cross-AZ ou Cross-Region.
- A Réplica de leitura pode ser promovida manualmente como uma instância de banco de dados independente.
- Read Replicas suportam implantações Multi-AZ.
- Você pode usar réplicas de leitura para fazer backups lógicos, se quiser armazenar os backups externamente no RDS.
- Você pode ter réplicas de leitura de réplicas de leitura.

- Read Replica ajuda a manter uma cópia dos bancos de dados em uma região diferente para recuperação de desastres.
- Você pode ter até cinco réplicas de leitura por mestre, cada uma com seu próprio ponto de extremidade DNS.
- Ao contrário de uma réplica em espera Multi-AZ, você pode se conectar a cada réplica de leitura e usá-las para escalonamento de leitura.

## Summary

Você pode usar Read Replicas com Multi-AZ como parte de uma estratégia de recuperação de desastres (DR) para seus bancos de dados de produção.

Um plano de DR bem projetado e testado é fundamental para manter a continuidade dos negócios após um desastre.

Uma réplica de leitura em uma região diferente do banco de dados de origem pode ser usada como banco de dados de reserva e promovida para se tornar o novo banco de dados de produção em caso de interrupção regional.

Importante: as implantações Multi-AZ não são uma solução de escalonamento de leitura, você não pode usar uma réplica em espera para atender ao tráfego de leitura. O modo de espera está lá apenas para failover.

## ➤ Ponto de Atenção

1. Você está projetando uma arquitetura de aplicativo da web multicamadas que consiste em um conjunto de instâncias EC2 e um servidor de banco de dados relacional Oracle. É necessário que o banco de dados esteja altamente disponível e que você tenha controle total sobre o sistema operacional subjacente. Qual serviço da AWS você usará para sua camada de banco de dados? R: Instâncias do Amazon EC2 com replicação de dados entre duas zonas de disponibilidade diferentes
2. Há muitas interrupções na zona de disponibilidade de sua instância de banco de dados RDS a ponto de você perder o acesso ao banco de dados. O que você poderia fazer para evitar a perda de acesso ao seu banco de dados caso esse evento aconteça novamente? R: Enabled Multi-AZ failover
3. Você está trabalhando para uma seguradora como arquiteto de soluções sênior. A empresa tem um aplicativo que processa milhares de dados de clientes armazenados em um banco de dados Amazon MySQL com configuração de implantações Multi-AZ para alta disponibilidade em caso de tempo de inatividade. Nos últimos dias, você notou uma tendência crescente de operações de leitura e gravação, o que está aumentando a latência das consultas em seu banco de dados. Você está planejando usar a instância de banco de dados de reserva para equilibrar as operações de leitura e gravação da instância primária. Ao executar sua instância primária do Amazon RDS como uma implantação Multi-AZ, você pode usar a instância em espera para operações de leitura e gravação? R: Não (somente read replicas)
4. Uma plataforma de negociação Forex, que frequentemente processa e armazena dados financeiros globais a cada minuto, é hospedada em seu data center local e usa um banco de dados Oracle. Devido a um problema recente de resfriamento em seu data center, a empresa precisa urgentemente migrar sua infraestrutura para AWS para melhorar o desempenho de seus aplicativos. Como arquiteto de soluções, você é responsável por garantir que o banco de dados seja migrado corretamente e permaneça disponível em caso de falha do servidor de banco de dados no futuro. Qual das alternativas a seguir é a solução mais adequada para atender ao requisito? R: Create an Oracle database in RDS with Multi-AZ deployments.



# Amazon Database Services



Amazon ElastiCache



Amazon DynamoDB

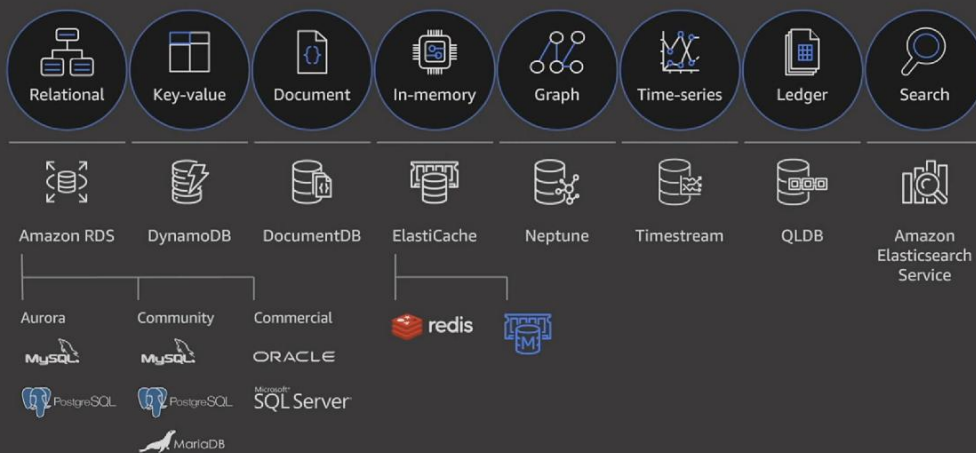


Redshift

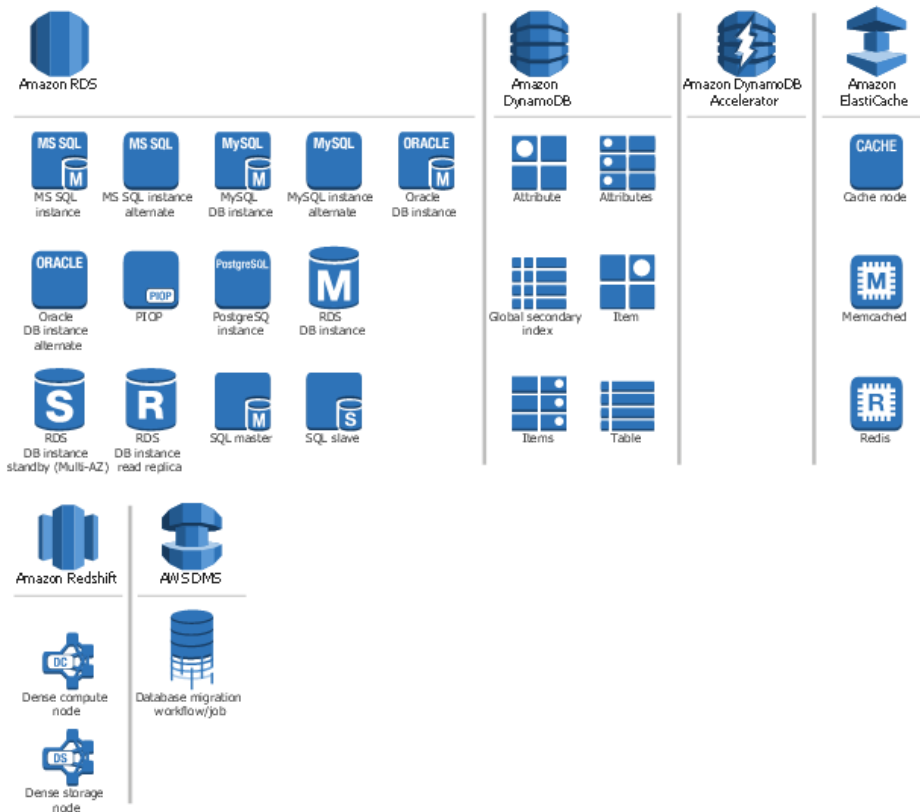


RDS

## Purpose-built databases



## Database



# Amazon Aurora

O Amazon Aurora é um banco de dados relacional compatível com MySQL e PostgreSQL criado para a nuvem. O Amazon Aurora é até cinco vezes mais rápido que bancos de dados MySQL padrão e três vezes mais rápido que bancos de dados PostgreSQL padrão.

O Amazon Aurora oferece um sistema de armazenamento distribuído, tolerante a falhas e com recuperação automática que escala automaticamente para até 128 TB por instância de banco de dados. O Amazon Aurora oferece altos níveis de performance e disponibilidade, com até 15 réplicas de leitura de baixa latência, recuperação point-in-time, backup contínuo para o Amazon S3 e replicação entre 3 zonas de disponibilidade (AZs).

O Amazon Aurora é um banco de dados de classe empresarial compatível com MySQL e PostgreSQL, começando em <\$1/day. Aurora supports up to 64TB of auto-scaling storage capacity, 6-way replication across three availability zones, and 15 low-latency read replicas.

## Características

Criado em 2014

Versão Proprietária da Amazon

Compatível com MySQL e PostgreSQL

5x mais rápido que outros databases

10x mais barato que outros databases

Inicia com 10 GB, AutoScaling é automático (64 GB em 64 GB)

Um único database pode ter até 64 TB

Até 15 réplicas de leitura (mais replicas menor latência)

Recover – Point-in-time

Backup contínuo – é distribuído em no mínimo 3 zonas diferentes dentro de buckets S3

## Types of Aurora endpoints

Um terminal é representado como um URL específico do Aurora que contém um endereço de host e uma porta. Os seguintes tipos de endpoints estão disponíveis em um cluster Aurora DB.

### Cluster endpoint

Um ponto de extremidade de cluster (or *writer endpoint*) para um cluster de banco de dados Aurora se conecta à instância de banco de dados primária atual para esse cluster de banco de dados. Este ponto de extremidade é o único que pode executar operações de gravação, como instruções DDL. Por causa disso, o ponto de extremidade do cluster é aquele ao qual você se conecta quando configura um cluster pela primeira vez ou quando seu cluster contém apenas uma única instância de banco de dados.

Cada cluster de banco de dados Aurora tem um ponto de extremidade de cluster e uma instância de banco de dados primária. Você usa o ponto de extremidade do cluster para todas as operações de gravação no cluster de banco de dados, incluindo inserções, atualizações, exclusões e alterações DDL. Você também pode usar o ponto de extremidade do cluster para operações de leitura, como queries.

O ponto de extremidade do cluster fornece suporte de failover para conexões de leitura/gravação com o cluster de banco de dados. Se a instância de banco de dados primária atual de um cluster de banco de dados falhar, o Aurora fará failover automaticamente para uma nova instância de banco de dados primária. Durante um failover, o cluster de banco de dados continua atendendo às solicitações de conexão ao ponto de extremidade do cluster da nova instância de banco de dados primária, com interrupção mínima do serviço. O exemplo a seguir ilustra um ponto de extremidade de cluster para um cluster Aurora MySQL DB.

```
mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com:3306
```

#### Reader endpoint

Um ponto de extremidade de leitor para um cluster de banco de dados Aurora fornece suporte de balanceamento de carga para conexões somente leitura com o cluster de banco de dados. Use o ponto de extremidade do leitor para operações de leitura, como consultas. Ao processar essas instruções nas réplicas Aurora somente leitura, este ponto de extremidade reduz a sobrecarga na instância primária. Também ajuda o cluster a dimensionar a capacidade de lidar com consultas SELECT simultâneas, proporcional ao número de réplicas Aurora no cluster. Cada cluster Aurora DB tem um terminal de leitor.

```
mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com:3306
```

#### Custom endpoint

Um ponto de extremidade personalizado para um cluster Aurora representa um conjunto de instâncias de banco de dados que você escolher. Quando você se conecta ao terminal, o Aurora executa o balanceamento de carga e escolhe uma das instâncias do grupo para lidar com a conexão. Você define a quais instâncias este terminal se refere e decide a que propósito o terminal serve.

Um cluster Aurora DB não tem pontos de extremidade personalizados até que você crie um. Você pode criar até cinco endpoints personalizados para cada cluster Aurora provisionado. Você não pode usar pontos de extremidade personalizados para clusters sem servidor Aurora. O ponto de extremidade personalizado fornece conexões de banco de dados com balanceamento de carga com base em critérios diferentes do recurso somente leitura ou leitura/gravação das instâncias de banco de dados.

```
myendpoint.cluster-custom-123456789012.us-east-1.rds.amazonaws.com:3306
```

#### Instance endpoint

Um ponto de extremidade de instância se conecta a uma instância de banco de dados específica dentro de um cluster Aurora. Cada instância de banco de dados em um cluster de banco de dados tem seu próprio ponto de extremidade de instância exclusivo. Portanto, há um ponto de extremidade de instância para a instância de banco de dados primária atual do cluster de banco de dados e há um ponto de extremidade de instância para cada uma das réplicas Aurora no cluster de banco de dados.

```
mydbinstance.123456789012.us-east-1.rds.amazonaws.com:3306
```

### ➤ **Pontos de Atenção**

1. Uma plataforma de compras online é hospedada em um grupo de Auto Scaling de instâncias Spot EC2 e usa Amazon Aurora PostgreSQL como seu banco de dados. Há um requisito para otimizar suas cargas de trabalho de banco de dados em seu cluster, onde você deve direcionar o tráfego de produção para suas instâncias de alta capacidade e apontar as consultas de relatórios enviadas por sua equipe interna para as instâncias de baixa capacidade. Qual é a configuração mais adequada para sua aplicação, bem como para seu cluster de banco de dados Aurora, para atender a esse requisito? R: Crie um novo endpoint personalizado no Aurora que balanceará a carga das conexões do banco de dados com base nos critérios especificados. Configure seu aplicativo para usar o endpoint personalizado para o tráfego de produção e as consultas de relatório.
2. A top university has recently launched its online learning portal where the students can take e-learning courses from the comforts of their homes. The portal is on a large On-Demand EC2 instance with a single Amazon Aurora database. How can you improve the availability of your Aurora database to prevent any unnecessary downtime of the online portal? R: Create Amazon Aurora Replicas.

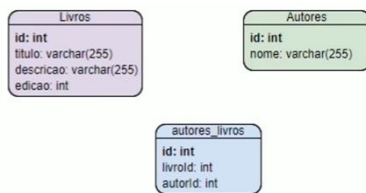
# Amazon DynamoDB

## NoSQL x SQL

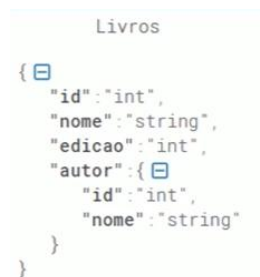
NoSQL é um tipo de banco de dados que possui esquemas flexíveis para a criação de aplicativos modernos. Eles são amplamente reconhecidos por sua facilidade de desenvolvimento, funcionalidade e performance em escala

Eles são otimizados especificamente para aplicativos que exigem modelos de grande volume de dados, baixa latência e flexibilidade. Esses requisitos são atendidos mediante o relaxamento de algumas restrições de consistência de dados dos outros bancos

Banco de dados SQL (Relacional) o modelo é projetado para permitir que o banco de dados imponha a integridade referencial entre as tabelas no banco de dados, normalizadas para reduzir a redundância e geralmente otimizadas para armazenamento



Banco de dados NoSQL (Não Relacional), neste modelo os dados são otimizados para desenvolvimento intuitivo e escalabilidade horizontal



Os bancos de dados NoSQL são ideias para muitos aplicativos modernos, como dispositivos móveis, Web e jogos que exigem banco de dados flexíveis, escaláveis, de alta performance e altamente funcionais para proporcionar ótimas experiências aos usuários

Fornecem esquemas flexíveis que permitem um desenvolvimento mais rápido e iterativo. O modelo de dados flexível torna os bancos de dados NoSQL ideias para dados semiestruturados e não estruturados

Escalabilidade: São projetados para serem escalados horizontalmente usando clusters distribuídos de hardware, em vez de escalá-los verticalmente adicionando servidores caros e robustos. Alguns provedores de nuvem lidam com essas operações nos bastidores como um serviço totalmente gerenciado

Alta Performance: Otimizados para modelos de dados específicos e padrões de acesso que permitem maior performance do que quando se tenta realizar uma funcionalidade semelhante com bancos de dados relacionais

Altamente Funcionais: Fornecem APIs e tipos de dados altamente funcionais criados especificamente para cada um de seus respectivos modelos de dados

Eles são altamente particionáveis e permitem escalabilidade horizontal em escalas que outros tipos de banco de dados não conseguem alcançar

O Amazon DynamoDB foi projetado para proporcionar uma latência consistente de um dígito de milissegundo para qualquer escala de cargas de trabalho

## AWS DynamoDB

Além de ser NoSQL e os dados serem acessados através de **chave/valor**, é um banco de dados durável, totalmente gerenciado com segurança, backup e restauração integrados, além de armazenamento em cache

O DynamoDB pode processar mais de 10 trilhões de solicitações por dia e comportar picos de mais de 20 milhões de solicitações por segundo. Empresas como Lyft, Airbnb, Samsung, Toyota, Netflix e muitas outras o utilizam

Performance em grande escala: Ele garante tempos de resposta consistente abaixo de **10 milissegundos**. As **tabelas globais** do DynamoDB replicam seus dados em várias regiões da AWS para oferecer acesso rápido e local a dados para seus aplicativos distribuídos globalmente

Ainda mais performance: Para casos de uso que exigem acesso ainda mais rápido com latência de microssegundos, o **DynamoDB Accelerator (DAX)** oferece um cache de memória totalmente gerenciado

Não há servidores para gerenciar. O DynamoDB é **sem servidor**, e é capaz de expandir e reduzir tabelas automaticamente para ajustar de acordo com a capacidade e manter o desempenho. A disponibilidade e a tolerância a falhas são incorporadas, eliminando a necessidade de projetar esses recursos em seus aplicativos

Modos de Capacidade: O DynamoDB oferece modos de **capacidade provisionada** e **sob demanda** para que se possa otimizar custos especificando a capacidade por carga de trabalho ou pagamento somente pelos recursos que consumir, lembrando que a capacidade sob demanda é um pouco mais cara que a capacidade provisionada

Pronto para uso empresarial: O DynamoDB **criptografa todos os dados por padrão** e oferece controle refinado de acesso e identidade em todas as suas tabelas. É possível criar **backups completos** de centenas de terabytes de dados instantaneamente, sem impacto no desempenho de suas tabelas, e recuperar qualquer momento dos 35 dias anteriores sem tempo de inatividade

## Armazenamento

Armazenamento é feito em SSD

Armazenado em no mínimo 3 Data Centers

Consistent Read – Leitura maior que 1 seg.

Strongly Consistent Reads – Leitura menor que 1 seg.

## Mais Sobre AWS DynamoDB

Latência em milissegundos

Unidade de Capacidade de Leitura (RCU): Unidade responsável por ler registros do banco de dados

Unidade de Capacidade de Gravação (WCU): Unidade responsável por gravar registros do banco de dados

Cada **chamada de API** para ler dados de uma tabela é uma solicitação de leitura

As unidades de leitura do DynamoDB podem ser **eventualmente consistentes**, **fortemente consistentes** ou **transacionais**

Eventualmente Consistente: Quando se lê dados de uma tabela do DynamoDB, a resposta pode não refletir os resultados de uma operação de gravação recentemente concluída. A resposta pode incluir alguns dados obsoletos. Se você repetir sua solicitação de leitura após um curto período, a resposta deverá retornar os dados mais recentes

Fortemente Consistente: Quando é solicitada uma leitura fortemente consistente, o DynamoDB retorna uma resposta com os dados mais atualizados, refletindo as atualizações de todas as operações de gravação anteriores que foram bem-sucedidas. No entanto, essa consistência traz algumas desvantagens, tais como:

Uma leitura fortemente consistente **poderá não estar disponível** se houver um **atraso ou uma interrupção de rede**. Retornando um erro de servidor (HTTP 500). Podem também ter **maior latência** do que as eventualmente consistentes. E por fim, elas usam **maior capacidade de taxa de transferência** do que as leituras eventualmente consistentes e mais RCU



## Unidade de Capacidade de Leitura (RCU)

Para itens com tamanho de até 4KB, uma RCU pode executar uma solicitação de leitura fortemente consistente por segundo, itens maiores que 4KB exigem mais RCUs. Para itens com tamanho de até 4KB, uma RCU pode executar duas solicitações de leitura eventualmente consistente por segundo. As solicitações de leitura transacional exigem duas RCUs para executar uma leitura por segundo para itens de até 4KB. É possível escalar um número específico ou de acordo com a demanda

## Unidade de Capacidade de Gravação (WCU)

Cada chamada de API para gravar dados de uma tabela é uma solicitação de gravação. Uma unidade de solicitação de gravação padrão pode gravar um item de até 1KB/segundo. Para itens maiores que 1KB, são necessárias unidades de solicitação de gravação adicionais. Uma gravação transacional exige duas unidades de solicitação de gravação. É possível escalar um número específico ou de acordo com a demanda

## Solicitação de Leitura/Gravação Transacionais

No DynamoDB, uma leitura ou gravação transacional é diferente de uma leitura ou gravação padrão porque garante que **todas as operações de um único conjunto** de transações executem com sucesso ou falhem em conjunto

## Tabelas Globais e rWCU

As tabelas globais replicam automaticamente as tabelas do DynamoDB nas regiões da AWS à sua escolha. Cada gravação ocorre na **região local** e nas **regiões replicadas**. Unidade de Capacidade de Gravação replicadas (**rWCU**) é a unidade responsável por gravar registros do banco de dados nas tabelas replicadas, lembrando que a replicação ocorre quase que em tempo real, ou seja, gravou um dado localmente o registro é automaticamente replicado

## Streams do AWS DynamoDB

Um stream do DynamoDB é um fluxo ordenado de informações sobre alterações em itens de uma tabela do DynamoDB. Quando é ativado um stream em uma tabela, o DynamoDB captura informações sobre todas as modificações em itens de dados na tabela e pode por exemplo invocar um lambda, ou seja, as inserções, alterações e remoções na tabela são interpretados como eventos que podem disparar um lambda

## Definição de Preço para Capacidade Provisionada

Com o modo de capacidade provisionada, você especifica o número de leituras e gravações de dados por segundo exigidas pelo aplicativo. Você pode usar o **Auto Scaling** para ajustar automaticamente a capacidade da tabela de acordo com a taxa de utilização do aplicativo e reduzir os custos. Com o Auto Scaling, é possível configurar o mínimo e máximo de unidades de leitura e gravação e o DynamoDB escala conforme a necessidade e quantidade de requisições

## Definição de Preço para Capacidade Sob Demanda

Com o modo de capacidade sob demanda, você paga pelas leituras e gravações de dados efetuadas pelo aplicativo nas tabelas. Não é preciso especificar a **quantidade de throughput** (taxa de transferência) de leitura e gravação que espera que o aplicativo execute, pois o DynamoDB acomoda instantaneamente o aumento e a redução das cargas de trabalho, lembrando que o preço pode ficar mais elevado de acordo com a necessidade

## Cardinalidade

Em modelagem de dados a cardinalidade é um dos princípios fundamentais sobre relacionamento de um banco de dados relacional. Nela são definidos o graus de relação entre duas entidades ou tabelas. No modelo relacional, podemos ter os seguintes níveis de relacionamento: 1:N, N:N, 1:1.

Alta cardinalidade significa que há vários valores únicos (como em um campo para números de telefone).

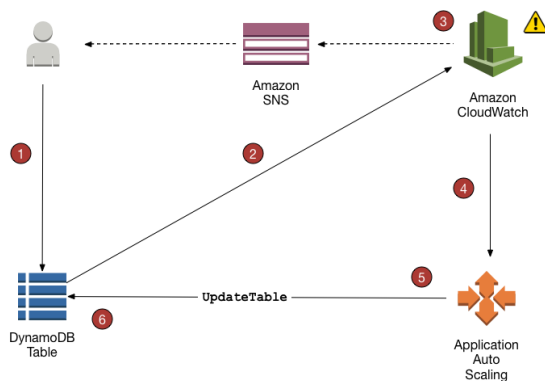
A cardinalidade entre as tabelas é semelhante, mas refere-se a se uma linha de uma tabela poderia ser vinculada a mais de uma linha em outra tabela.

## Managing Throughput Capacity Automatically with DynamoDB Auto Scaling

If your application can handle some throttling from DynamoDB when there's a sudden increase in traffic, you should use DynamoDB Auto Scaling.

Amazon DynamoDB auto scaling uses the AWS Application Auto Scaling service to dynamically adjust provisioned throughput capacity on your behalf, in response to actual traffic patterns. This enables a table or a global secondary index to increase its provisioned read and write capacity to handle sudden increases in traffic, without throttling. When the workload decreases, Application Auto Scaling decreases the throughput so that you don't pay for unused provisioned capacity.

If you use the AWS Management Console to create a table or a global secondary index, DynamoDB auto scaling is enabled by default. You can modify your auto scaling settings at any time.



## DynamoDB stream

A *DynamoDB stream* is an ordered flow of information about changes to items in an Amazon DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

When you enable DynamoDB Streams on a table, you can associate the stream ARN with a Lambda function that you write. Immediately after an item in the table is modified, a new record appears in the table's stream. AWS Lambda polls the stream and invokes your Lambda function synchronously when it detects new stream records. Since we require an immediate entry made to an application in case an item in the DynamoDB table is modified, a lambda function is also required.

Let us try to analyze this with an example:

Consider a mobile gaming app that writes to a GamesScores table. Whenever the top score of the Game Scores table is updated, a corresponding stream record is written to the table's stream. This event could trigger a Lambda function that posts a Congratulatory message on a Social media network handle.

DynamoDB streams can be used to monitor the changes to a DynamoDB table.

## Links Úteis

[https://docs.aws.amazon.com/pt\\_br/sdk-for-javascript/v2/developer-guide/dynamodb-example-table-read-write.html](https://docs.aws.amazon.com/pt_br/sdk-for-javascript/v2/developer-guide/dynamodb-example-table-read-write.html)

[https://docs.aws.amazon.com/pt\\_br/sdk-for-javascript/v2/developer-guide/dynamodb-example-document-client.html](https://docs.aws.amazon.com/pt_br/sdk-for-javascript/v2/developer-guide/dynamodb-example-document-client.html)

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted.NodeJs.04.html>

<https://docs.aws.amazon.com/lambda/latest/dg/kinesis-tutorial-spec.html>

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/AutoScaling.html>

## ➤ Pontos de Atenção

1. Instruído a construir uma arquitetura da web usando instâncias do EC2 On-Demand e um banco de dados na AWS. No entanto, devido a restrições de orçamento, a empresa instruiu você a escolher um serviço de banco de dados no qual eles não precisassem mais se preocupar com tarefas de gerenciamento de banco de dados, como provisionamento de hardware ou software, instalação, configuração, dimensionamento e backups. Qual serviço de banco de dados na AWS é melhor para usar neste cenário? R: DynamoDB
2. Um aplicativo Docker, que está sendo executado em um cluster Amazon ECS atrás de um balanceador de carga, usa intensamente o DynamoDB. Você é instruído a melhorar o desempenho do banco de dados distribuindo a carga de trabalho uniformemente e usando o rendimento provisionado com eficiência. Qual das opções a seguir você consideraria implementar para sua tabela do DynamoDB? R: Use chaves de partição com atributos de alta cardinalidade, que possuem um grande número de valores distintos para cada item.
3. Your manager has asked you to deploy a mobile application that can collect votes for a popular singing competition. Millions of users from around the world will submit votes using their mobile phones. These votes must be collected and stored in a highly scalable and highly available data store which will be queried for real-time ranking. Which of the following combination of services should you use to meet this requirement? R: Amazon DynamoDB and AWS AppSync
4. A popular augmented reality (AR) mobile game is heavily using a RESTful API which is hosted in AWS. The API uses Amazon API Gateway and a DynamoDB table with a preconfigured read and write capacity. Based on your systems monitoring, the DynamoDB table begins to throttle requests during high peak loads which causes the slow performance of the game. Which of the following can you do to improve the performance of your app? R: Add the DynamoDB table to an Auto Scaling Group.
5. A leading IT consulting company has an application which processes a large stream of financial data by an Amazon ECS Cluster then stores the result to a DynamoDB table. You have to design a solution to detect new entries in the DynamoDB table then automatically trigger a Lambda function to run some tests to verify the processed data. What solution can be easily implemented to alert the Lambda function of new entries while requiring minimal configuration change to your architecture? R: Enable DynamoDB Streams to capture table activity and automatically trigger the Lambda function.
6. A popular mobile game uses CloudFront, Lambda, and DynamoDB for its backend services. The player data is persisted on a DynamoDB table and the static assets are distributed by CloudFront. However, there are a lot of complaints that saving and retrieving player information is taking a lot of time. To improve the game's performance, which AWS service can you use to reduce DynamoDB response times from milliseconds to microseconds? R: Amazon DynamoDB Accelerator (DAX)
7. Your company currently has setup their data store on AWS DynamoDB. One of your main revenue generating applications uses the tables in this service. Your application is now expanding to 2 different other locations and you want to ensure that the latency for data retrieval is the least from the new regions. Which of the following can help accomplish this? R: Enable global tables for DynamoDB
8. Uma empresa configurou um aplicativo no AWS que interage com o DynamoDB. É necessário que, quando um item for modificado em uma tabela do DynamoDB, seja feita uma entrada imediata no aplicativo associado. Como pode ser isto alcançado? R: 1. Use DynamoDB streams to monitor the changes to the DynamoDB table. 2. Trigger a lambda function to make an associated entry in the application as soon as the DynamoDB streams are modified.
9. Sua equipe foi instruída a desenvolver um aplicativo que fará uso de uma tabela DynamoDB. Durante o estágio de design, você deve fornecer entradas para garantir que uma estratégia ideal seja empregada para uma alta

expectativa de leitura e gravação na tabela do DynamoDB subjacente. Qual das opções a seguir você consideraria? R: Use partition keys with a large number of distinct values for the underlying table

# Amazon ElastiCache

Armazenamento de dados na memória totalmente gerenciado, compatível com Redis ou Memcached. Disponibilize latência inferior a um milissegundo para aplicações em tempo real.

Com o Amazon ElastiCache, você pode configurar, executar e escalar sem problemas conhecidos repositórios de dados na memória compatíveis com código aberto na nuvem.

Crie aplicações com uso intenso de dados ou aumente a performance de aplicativos existentes recuperando dados de armazenamentos de dados na memória de alta taxa de transferência e baixa latência.

O Amazon ElastiCache é uma opção popular para casos de uso em tempo real, como armazenamento em cache, armazenamentos de sessão, jogos, serviços geoespaciais, análises em tempo real e filas.

Não será mais necessário executar tarefas de gerenciamento, como provisionamento de hardware, patching de software, configuração, monitoramento, recuperação de falhas e backups. O ElastiCache monitora continuamente os clusters para manter as cargas de trabalho em operação.

## Características

In Memory Cache: não se utiliza disco para guardar dados

## Tipos

Memcached: Key-value, Objetos

Redis: Key-value (Multi-AZ), registros

*Ao decidir se usar Redis ou Memcached, a principal diferença entre os dois é a persistência de dados. Enquanto o Redis é um armazenamento de dados na memória (principalmente) e não é volátil, o Memcached é um cache na memória e é volátil*

## Amazon ElastiCache for Redis

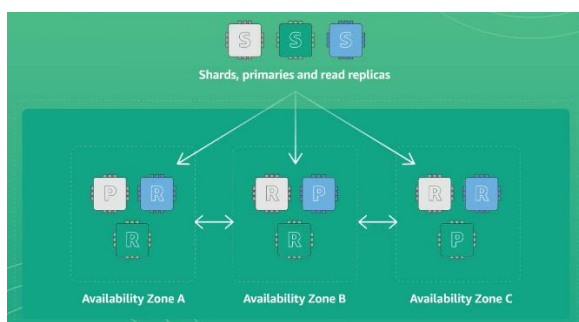
O Redis é um armazenamento de estrutura de dados de **chave-valor** de código aberto e na memória.

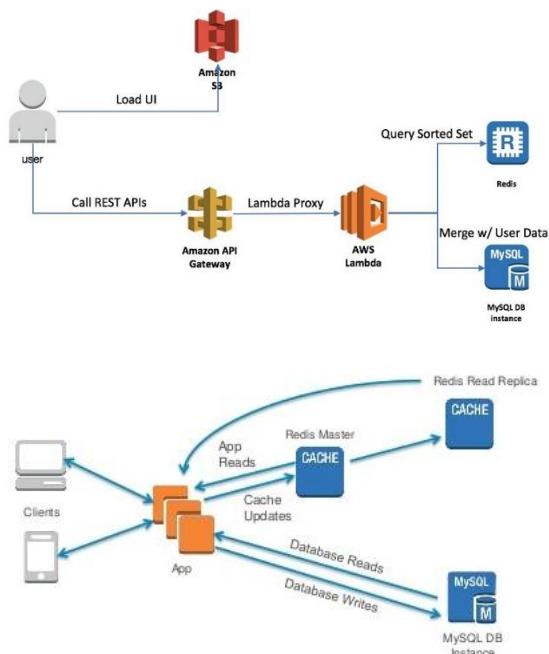
O Redis oferece um conjunto de estruturas versáteis de dados na memória que permite a fácil criação de várias aplicações personalizadas.

Com o Amazon ElastiCache for Redis, você pode começar aos poucos e escalar facilmente os dados do Redis à medida que a sua aplicação cresce, até chegar a um cluster com 340 TB de dados em memória.

Um ambiente de cluster do Redis pode ser escalado para até 500 nós e 500 estilhaços. Você pode redimensionar clusters online para aumentar ou diminuir a escala horizontal de clusters do Redis, sem tempo de inatividade e de acordo com a evolução de demanda.

Para escalar a capacidade de leitura, o ElastiCache permite adicionar até cinco réplicas de leitura em várias zonas de disponibilidade. Para escalar a capacidade de gravação, o ElastiCache oferece suporte ao modo de cluster Redis, que permite particionar o tráfego de gravação em vários primários.





## Amazon ElastiCache para Memcached

Memcached é um sistema de armazenamento de objetos em cache de memória largamente adotado.

O Memcached é um armazenamento conhecido de **chave-valor na memória** distribuído usado para criar esses aplicativos. Alguns dos casos de uso comuns do Memcached incluem armazenamento em cache de dados de referência, armazenamento em cache de consulta de banco de dados e, em alguns casos, como armazenamento de sessão.

O Amazon ElastiCache com Memcached inclui fragmentação para escalar o cache na memória com até 20 nós e 12,7 TiB por cluster.

O cliente conecta-se a um único endpoint de cluster e obtém informações de configuração de outros nós. Quando os nós são adicionados ou removidos, o cliente reconfigura-se para usar o maior conjunto atual de nós.

O Amazon ElastiCache para Memcached oferece suporte à Amazon VPC e permite que você isole o cluster nos intervalos IP escolhidos para os nós e use-os para conectar-se ao aplicativo. O ElastiCache monitora continuamente seus nós e aplica os patches necessários para manter seu ambiente seguro.

## Authenticating users with the Redis AUTH command

Redis authentication tokens, or passwords, enable Redis to require a password before allowing clients to run commands, thereby improving data security.

You can require that users enter a token (password) on a token-protected Redis server. To do this, include the parameter `--auth-token` (API: `AuthToken`) with the correct token when you create your replication group or cluster. Also include it in all subsequent commands to the replication group or cluster.

### Key parameters

- `--engine` – Must be redis.
- `--engine-version` – Must be 3.2.6, 4.0.10, or later.
- `--transit-encryption-enabled` – Required for authentication and HIPAA eligibility.
- `--auth-token` – Required for HIPAA eligibility. This value must be the correct token for this token-protected Redis server.
- `--cache-subnet-group` – Required for HIPAA eligibility.

## Data security in Amazon ElastiCache

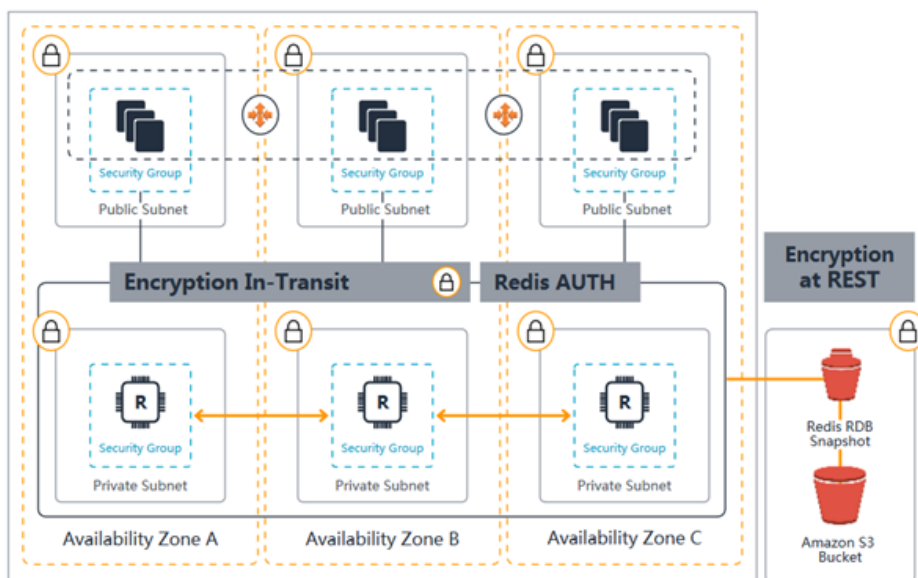
To help keep your data secure, Amazon ElastiCache and Amazon EC2 provide mechanisms to guard against unauthorized access of your data on the server.

Amazon ElastiCache for Redis also provides optional encryption features for data on clusters running Redis versions 3.2.6, 4.0.10 or later:

- In-transit encryption encrypts your data whenever it is moving from one place to another, such as between nodes in your cluster or between your cluster and your application.
- At-rest encryption encrypts your on-disk data during sync and backup operations.

If you want to enable in-transit or at-rest encryption, you must meet the following conditions.

- Your cluster or replication group must be running Redis 3.2.6, 4.0.10 or later.
- Your cluster or replication group must be created in a VPC based on Amazon VPC.
- Optionally, you can also use AUTH and the AUTH token (password) needed to perform operations on this cluster or replication group.



*ElastiCache for Redis Security Diagram*

## Session Management

A centralized session management data store provides consistent user experience, better session durability, and high availability

### Session Management

Existem várias maneiras de gerenciar as sessões de usuário, incluindo o armazenamento dessas sessões localmente no nó que responde à solicitação HTTP ou designando uma camada em sua arquitetura que pode armazenar essas sessões de maneira escalável e robusta. As abordagens comuns usadas incluem o uso de sessões Sticky ou de Cache Distribuído para o gerenciamento de sessão. Essas abordagens são descritas a seguir.

### Sticky Sessions with Local Session Caching

Sticky sessions, also known as session affinity, allow you to route a site user to the particular web server that is managing that individual user's session. The session's validity can be determined by a number of methods, including

a client-side cookies or via configurable duration parameters that can be set at the load balancer which routes requests to the web servers.

Some advantages with utilizing sticky sessions are that it's cost effective due to the fact you are storing sessions on the same web servers running your applications and that retrieval of those sessions is generally fast because it eliminates network latency. A drawback for using storing sessions on an individual node is that in the event of a failure, you are likely to lose the sessions that were resident on the failed node. In addition, in the event the number of your web servers change, for example a scale-up scenario, it's possible that the traffic may be unequally spread across the web servers as active sessions may exist on particular servers. If not mitigated properly, this can hinder the scalability of your applications.

## Distributed Session Management

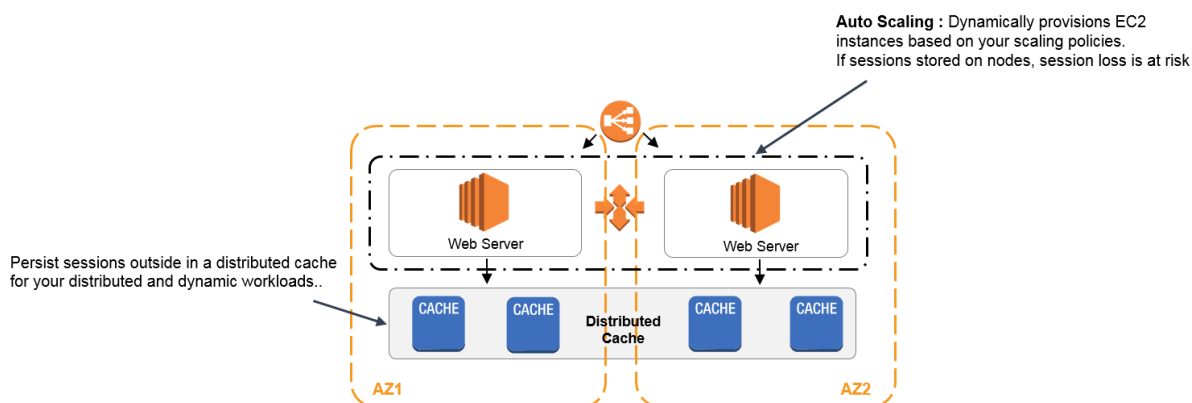
In order to address scalability and to provide a shared data storage for sessions that can be accessible from any individual web server, you can abstract the HTTP sessions from the web servers themselves. A common solution to for this is to leverage an In-Memory Key/Value store such as Redis and Memcached.

While Key/Value data stores are known to be extremely fast and provide sub-millisecond latency, the added network latency and added cost are the drawbacks. An added benefit of leveraging Key/Value stores is that they can also be utilized to cache any data, not just HTTP sessions, which can help boost the overall performance of your applications.

A consideration when choosing a distributed cache for session management is determining how many nodes may be needed in order to manage the user sessions. Generally speaking, this decision can be determined by how much traffic is expected and/or how much risk is acceptable. In a distributed session cache, the sessions are divided by the number of nodes in the cache cluster. In the event of a failure, only the sessions that are stored on the failed node are affected. If reducing risk is more important than cost, adding additional nodes to further reduce the percent of stored sessions on each node may be ideal even when fewer nodes are sufficient.

Another consideration may be whether or not the sessions need to be replicated or not. Some key/value stores offer replication via read replicas. In the event of a node failure, the sessions would not be entirely lost. Whether replica nodes are important in your individual architecture may inform as to which key/value store should be used. ElastiCache offerings for In-Memory key/value stores include ElastiCache for Redis, which can support replication, and ElastiCache for Memcached which does not support replication.

There are a number of ways to store sessions in Key/Value stores. Many application frameworks provide libraries which can abstract some of the integration plumbing required to GET/SET those sessions in memory. In other cases, you can write your own session handler to persist the sessions directly.





## Links Úteis

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/auth.html>

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/encryption.html>

<https://aws.amazon.com/caching/session-management/>

### ➤ Pontos de Atenção

1. Seu aplicativo da web depende inteiramente de bancos de dados baseados em disco mais lentos, fazendo com que seu desempenho seja lento. Para melhorar seu desempenho, você integrou um armazenamento de dados na memória ao seu aplicativo da web usando o ElastiCache. Como o Amazon ElastiCache melhora o desempenho do banco de dados? R: Armazenando em cache os resultados da consulta do banco de dados.
2. Você está projetando um portal bancário que usa o Amazon ElastiCache para Redis como seu componente de gerenciamento de sessão distribuída. Como os outros engenheiros de nuvem em seu departamento têm acesso ao cluster ElastiCache, você deve proteger os dados da sessão no portal exigindo que eles insiram uma senha antes de receberem permissão para executar comandos do Redis. Como arquiteto de soluções, qual das seguintes opções você deve fazer para atender ao requisito acima? R: Authenticate the users using Redis AUTH by creating a new Redis Cluster with both the --transit-encryption-enabled and --auth-token parameters enabled.
3. You are designing an online banking application which needs to have a distributed session data management. Currently, the application is hosted on an Auto Scaling group of On-Demand EC2 instances across multiple Availability Zones with a Classic Load Balancer that distributes the load. Which of the following options should you do to satisfy the given requirement? R: Use Amazon ElastiCache
4. You have a fleet of running Spot EC2 instances behind an Application Load Balancer. The incoming traffic comes from various users across multiple AWS regions and you would like to have the user's session shared among your fleet of instances. You are required to set up a distributed session management layer that will provide a scalable and shared data storage for the user sessions. Which of the following would be the best choice to meet the requirement while still providing sub-millisecond latency for your users? R: ElastiCache in-memory caching

# Amazon RDS

O Amazon RDS gerencia tarefas administrativas complexas e demoradas, como instalação e atualização do software PostgreSQL, gerenciamento do armazenamento, replicação para alta disponibilidade e throughput de leitura, e backups para recuperação de desastres.

O Amazon RDS está disponível em vários tipos de instância de banco de dados – com otimização para memória, performance ou E/S – e oferece seis mecanismos de bancos de dados comuns, incluindo Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database e SQL Server. Você pode usar o AWS Database Migration Service para migrar ou replicar facilmente bancos de dados existentes para o Amazon RDS.

## Amazon Aurora

O Amazon Aurora é um banco de dados relacional compatível com MySQL e PostgreSQL criado para a nuvem. O Amazon Aurora é até cinco vezes mais rápido que bancos de dados MySQL padrão e três vezes mais rápido que bancos de dados PostgreSQL padrão.

O Amazon Aurora oferece um sistema de armazenamento distribuído, tolerante a falhas e com recuperação automática que escala automaticamente para até 128 TB por instância de banco de dados. O Amazon Aurora oferece altos níveis de performance e disponibilidade, com até 15 réplicas de leitura de baixa latência, recuperação point-in-time, backup contínuo para o Amazon S3 e replicação entre 3 zonas de disponibilidade (AZs).

O Amazon Aurora é um banco de dados de classe empresarial compatível com MySQL e PostgreSQL, começando em <\$1/day. Aurora supports up to 64TB of auto-scaling storage capacity, 6-way replication across three availability zones, and 15 low-latency read replicas.

## Oracle Database

Você pode usar instâncias de banco de dados executando Oracle dentro de um VPC. Você também pode adicionar recursos à sua instância de banco de dados Oracle, habilitando várias opções. Amazon RDS oferece suporte a implantações Multi-AZ para Oracle como uma solução de failover de alta disponibilidade

*Failover: Em computação, tolerância a falhas é a comutação para um computador servidor, sistema, componente de hardware ou rede redundante ou em modo de espera em caso de falha ou finalização anormal daquele ativo previamente.*

Para oferecer uma experiência de serviço gerenciado, o Amazon RDS não fornece acesso shell a instâncias de banco de dados. Ele também restringe o acesso a certos procedimentos e tabelas do sistema que precisam de privilégios avançados. Você pode acessar seu banco de dados usando clientes SQL padrão, como Oracle SQL \* Plus. No entanto, você não pode acessar o host diretamente usando Telnet ou Secure Shell (SSH).

## Tópicos Importantes

Instance classes, storage, and PIOPS: Se você estiver criando uma instância de banco de dados para fins de produção, deve compreender como as classes de instância, os tipos de armazenamento e o IOPS provisionado funcionam no Amazon RDS.

Multi-AZ deployments: Uma instância de banco de dados de produção deve usar implantações Multi-AZ. As implantações Multi-AZ fornecem maior disponibilidade, durabilidade de dados e tolerância a falhas para instâncias de banco de dados.

Amazon Virtual Private Cloud (VPC): Se sua conta da AWS tiver um VPC padrão, sua instância de banco de dados será criada automaticamente dentro do VPC padrão.

Security Groups: Por padrão, as instâncias de banco de dados são criadas com um firewall que impede o acesso a elas. Portanto, você deve criar um grupo de segurança com os endereços IP e configuração de rede corretos para acessar a instância de banco de dados. O grupo de segurança que você cria depende da plataforma do Amazon EC2

em que sua instância de banco de dados está e se você acessará sua instância de banco de dados a partir de uma instância do Amazon EC2.

Parameter Groups: Se sua instância de banco de dados vai exigir parâmetros de banco de dados específicos, você deve criar um grupo de parâmetros antes de criar a instância de banco de dados.

Connecting to your DB instance: Depois de criar um grupo de segurança e associá-lo a uma instância de banco de dados, você pode se conectar à instância de banco de dados usando qualquer aplicativo cliente SQL padrão, como o Microsoft SQL Server Management Studio.

Backup and restore: Ao criar sua instância de banco de dados, você pode configurá-la para fazer backups automatizados. Você também pode fazer backup e restaurar seus bancos de dados manualmente usando arquivos de backup completo (files .bak).

Log files: Você pode acessar os arquivos de log da sua instância de banco de dados do SQL Server.

## Backups

Automated: armazenamento de 1 a 35 dias, backups ficam armazenados em S3. Totalmente Gratuito

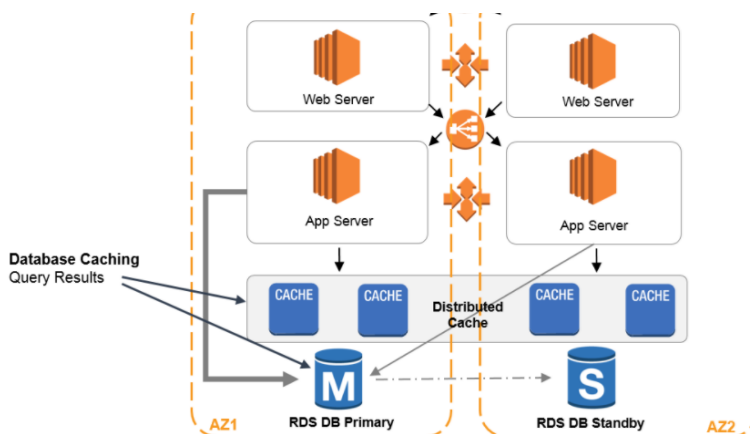
DB Snapshot: o backup é manual, um snapshot é criado naquele instante

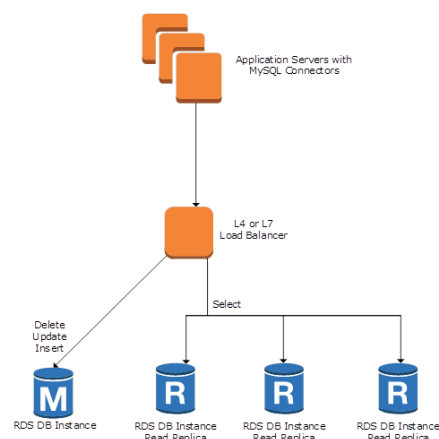
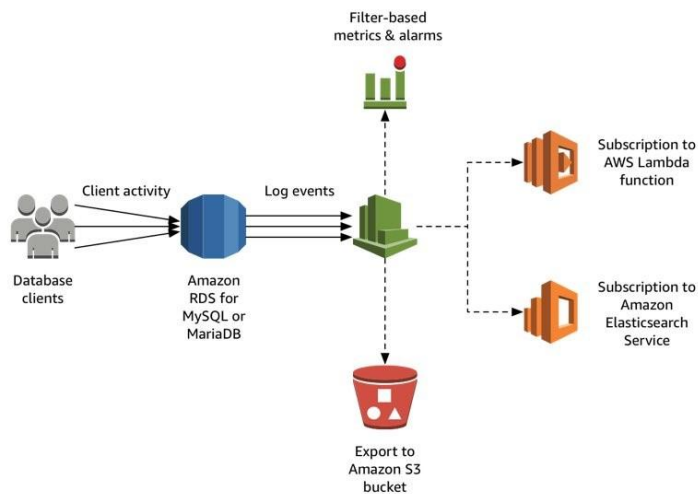
**Observação: para que a Read Replica funcione é obrigatório que o backup esteja habilitado**

## Read Replicas and ElastiCache

AWS says, "Amazon RDS Read Replicas provide enhanced performance and durability for database (DB) instances. **This feature makes it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads.** You can create one or more replicas of a given source DB Instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput".

Amazon ElastiCache is an in-memory cache that can be used to cache common read requests. The below diagram shows how caching can be added to an existing architecture:





## ➤ Pontos de Atenção

Um registro de Nome Canônico, do inglês *Canonical Name record (CNAME)*, é um tipo de registro de recurso no Sistema de Nomes de Domínio que mapeia um nome de domínio para outro. Isso pode ser conveniente ao executar vários serviços a partir de um único endereço IP.

1. A tech company is running two production web servers hosted on Reserved EC2 instances with EBS-backed root volumes. These instances have a consistent CPU load of 90%. Traffic is being distributed to these instances by an Elastic Load Balancer. In addition, they also have Multi-AZ RDS MySQL databases for their production, test, and development environments. What recommendation would you make to reduce cost in this AWS environment without affecting availability and performance of mission-critical systems? Choose the best answer. R: Consider not using a Multi-AZ RDS deployment for the development and test database
2. You are working as a Junior Solutions Architect where you are responsible in enhancing the availability and durability of the database instances in your VPC. Your company has a Multi-AZ RDS instance in the ap-northeast-1 region. If a storage volume on the primary instance fails in a Multi-AZ deployment, Amazon RDS automatically initiates a failover to the up-to-date standby instance. In case of a failover, which record in Route 53 is changed?  
R: CNAME
3. A grande empresa de manufatura para a qual você trabalha está interessada em mover seu parque de produção para a AWS. Eles executam uma loja Joomla que utiliza MySQL no back-end. Atualmente, eles também usam bancos de dados MySQL em cluster em uma configuração ativa / passiva em um único local. Ao mudar para a AWS, eles querem uma configuração ativa / passiva em 2 locais geograficamente distintos, com failover

automático entre os dois. Como arquiteto de soluções, qual das seguintes opções de RDS você deve recomendar? R: RDS Multi-AZ

4. A web application, which is hosted in your on-premises data center and uses a MySQL database, must be migrated to AWS Cloud. You need to ensure that the network traffic to and from your RDS database instance is encrypted using SSL. For improved security, you have to use the profile credentials specific to your EC2 instance to access your database, instead of a password. Which of the following should you do to meet the above requirement? R: **Set up an RDS database and enable the IAM DB Authentication.**
5. You have a customer hosting their website on a cluster of web servers behind a internet facing load balancer. The web application interfaces with an RDS database. The management team has specified that the database service continue to function even in the event of failures on the primary database server. A secondary database needs to be available as quickly as possible. Which would provide that capability best? R: Create a Multi-AZ database.
6. You are working in a Research and Development Department in IT Company where you as a Cloud Architect, trying to check the impact on real-time transactions. You created a multi-AZ RDS setup consisting of a Primary instance and a Read-replica. What is the impact of the read-replica on the transactions in the primary instance? R: Transactions are not impacted. For multi-AZ high availability, RDS uses synchronous replication between primary and standby systems. On the other hand, RDS Read Replica uses asynchronous replication, and any slowness in the Read Replica instance would simply cause data lag only in the read - replica. Transactions in primary are not impacted.
7. You are working as a Cloud Architect in a big IT Firm. To ensure high availability of both the web servers and your web application database, you deployed your auto-scaled EC2 instances in multiple Availability Zones with an Application Load Balancer in front. You configured Multi-Availability Zone to your RDS instance. There is a spike in incoming requests in the past few hours, and the performance of the primary database is starting to go down. What would happen to the database if the primary DB instance fails? R: The **canonical name (CNAME)** record is changed from the primary database to the standby database.
8. Você está trabalhando como consultor da AWS em uma organização de comércio eletrônico. A organização está planejando migrar seu banco de dados local para o Amazon RDS. O backup automatizado ajuda a restaurar o banco de dados a qualquer momento específico durante o período de retenção de backup no Amazon RDS. Qual ação foi executada como parte do processo de backup automatizado do Amazon RDS? R: **AWS creates a storage volume snapshot of the database instance during the backup window once a day. AWS RDS also captures transactions logs and uploads them to S3 buckets every 5 minutes.**
9. Uma auditoria de segurança descobre que uma de suas instâncias RDS MySQL não está criptografada. A instância tem uma réplica de leitura na mesma região da AWS que também não é criptografada. Você precisa corrigir esse problema o mais rápido possível. Qual é a maneira correta de adicionar criptografia à instância e sua réplica? R: **Create a DB snapshot from the instance. Copy the DB snapshot with encryption enabled. Restore a new DB instance from the new encrypted snapshot and configure a Read Replica in the new DB instance.**

# Amazon Redshift

*Um data lake ou lago de dados é um sistema ou repositório de dados armazenados em seu formato natural / bruto, geralmente objetos blobs ou arquivos. O data lake é um tipo de repositório que armazena conjuntos grandes e variados de dados brutos em formato nativo. Com os data lakes, você tem uma visão não refinada dos dados. Essa estratégia de gerenciamento é cada vez mais usada por empresas que querem um grande repositório holístico para armazenar dados.*

Com o Redshift, você pode consultar e combinar exabytes de dados estruturados e semiestruturados em data warehouses, bancos de dados operacionais e seu data lake usando o SQL padrão.

O Redshift permite salvar facilmente os resultados de suas consultas de volta no data lake do S3 usando formatos abertos, como o Apache Parquet, para que você possa fazer análises adicionais a partir de outros serviços analíticos como o Amazon EMR, o Amazon Athena e o Amazon SageMaker.

## Características

Os dados são armazenados e lidos no formato de Colunas

O Database inicia com **160 GB**

Sistema de compressão de dados

Single Node – 1 instância de database

Compute Node – Até 128 instâncias de database

**MPP (Massive Parallel Processing)** – Consegue realizar a leitura em vários databases ao mesmo tempo.

## Não é Multi-AZ

*Massivamente paralelo é o termo para usar um grande número de processadores de computador para executar simultaneamente um conjunto de cálculos coordenados em paralelo.*

## Amazon Redshift Snapshots

Snapshots are point-in-time backups of a cluster. There are two types of snapshots: *automated* and *manual*. Amazon Redshift stores these snapshots internally in Amazon S3 by using an encrypted Secure Sockets Layer (SSL) connection.

Amazon Redshift automatically takes incremental snapshots that track changes to the cluster since the previous automated snapshot. Automated snapshots retain all of the data required to restore a cluster from a snapshot. You can create a snapshot schedule to control when automated snapshots are taken, or you can take a manual snapshot any time.

When you restore from a snapshot, Amazon Redshift creates a new cluster and makes the new cluster available before all of the data is loaded, so you can begin querying the new cluster immediately. The cluster streams data on demand from the snapshot in response to active queries, then loads the remaining data in the background.

## Enhanced VPC routing in Amazon Redshift

Quando você usa o roteamento VPC aprimorado do Amazon Redshift, o Amazon Redshift força todo o tráfego COPY e UNLOAD entre seu cluster e seus repositórios de dados por meio de sua nuvem privada virtual (VPC) com base no serviço Amazon VPC. Ao usar o roteamento VPC aprimorado, você pode usar recursos VPC padrão, como grupos de segurança VPC, listas de controle de acesso à rede (ACLs), endpoints VPC, políticas de endpoint VPC, gateways de Internet e servidores Domain Name System (DNS), conforme descrito no Guia do usuário do Amazon VPC. Você usa

esses recursos para gerenciar rigidamente o fluxo de dados entre o cluster do Amazon Redshift e outros recursos. Ao usar o roteamento VPC aprimorado para rotear o tráfego por meio do VPC, você também pode usar os registros de fluxo do VPC para monitorar o tráfego COPY e UNLOAD.

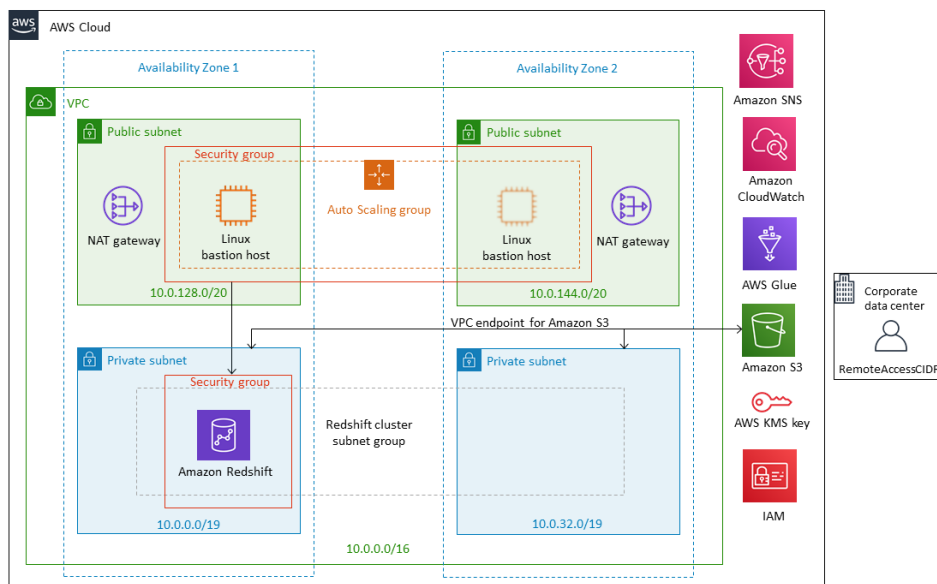
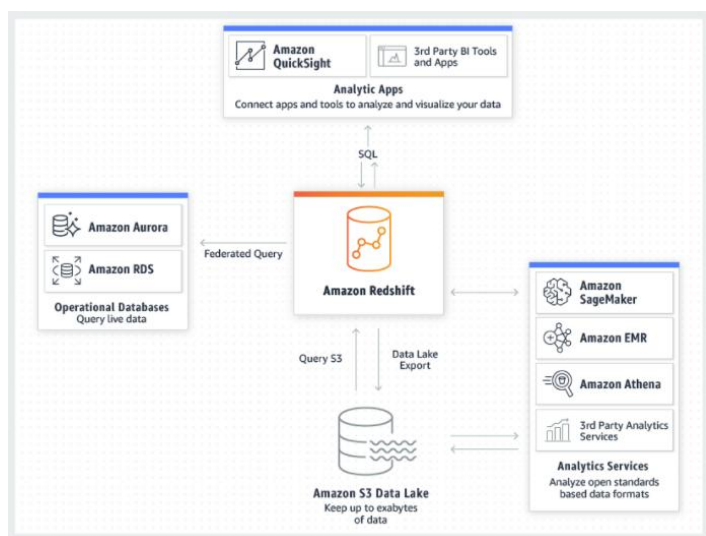
When you use Amazon Redshift Enhanced VPC Routing, Amazon Redshift forces all COPY and UNLOAD traffic between your cluster and your data repositories through your Amazon VPC.

If Enhanced VPC Routing is **not** enabled, Amazon Redshift routes traffic through the Internet, including traffic to other services within the AWS network.

## Links Úteis

<https://docs.aws.amazon.com/redshift/latest/mgmt/managing-snapshots-console.html>

<https://docs.aws.amazon.com/redshift/latest/mgmt/enhanced-vpc-routing.html>



## ➤ Pontos de Atenção

1. Você está trabalhando como consultor de TI para um grande banco de investimento que gera grandes conjuntos de dados financeiros com milhões de linhas. Os dados devem ser armazenados em colunas para reduzir o número de solicitações de E/S de disco e reduzir a quantidade de dados necessária para carregar do disco. O



banco possui um aplicativo de business intelligence de terceiros que se conecta ao serviço de armazenamento e, em seguida, gera relatórios financeiros diários e mensais para seus clientes em todo o mundo. Nesse cenário, qual é o melhor serviço de armazenamento a ser usado para atender ao requisito? R: Amazon Redshift

2. A data analytics company, which uses machine learning to collect and analyze consumer data, is using Redshift cluster as their data warehouse. You are instructed to implement a disaster recovery plan for their systems to ensure business continuity even in the event of an AWS region outage. Which of the following is the best approach to meet this requirement? **R: Enable Cross-Region Snapshots Copy in your Amazon Redshift Cluster.**
3. A company currently hosts a Redshift cluster in AWS. It should ensure that all traffic from and to the Redshift cluster does not go through the Internet for security reasons. Which features can be used to fulfill this requirement in an efficient manner? **R: Enable Amazon Redshift Enhanced VPC Routing**
4. Uma empresa atualmente hospeda um cluster Redshift na AWS. Deve garantir que todo o tráfego de e para o cluster Redshift não passe pela Internet por motivos de segurança. Quais recursos podem ser usados para atender a esse requisito de maneira eficiente? **R: Enable Amazon Redshift Enhanced VPC Routing.** When you use Amazon Redshift Enhanced VPC Routing, Amazon Redshift forces all COPY and UNLOAD traffic between your cluster and your data repositories through your Amazon VPC.
5. Uma empresa está planejando usar o serviço AWS Redshift. O serviço Redshift e seus dados seriam usados continuamente pelos próximos 3 anos, de acordo com o plano de negócios atual. Qual seria a solução mais econômica neste cenário? **R: Consider using Reserved Instances for the Redshift Cluster.** *If you intend to keep your Amazon Redshift cluster running continuously for a prolonged period, you should consider purchasing reserved node offerings. These offerings provide significant savings over on-demand pricing, but they require you to reserve compute nodes and commit to paying for those nodes for either a one-year or three-year duration.*
6. Uma empresa atualmente está hospedando um cluster Redshift. É necessário que os dados estejam disponíveis no cluster Redshift em outra região como parte da configuração de recuperação de desastre da empresa. Como você faria isso? **R: Configure cross-region snapshots for the underlying Redshift cluster.**

You can configure Amazon Redshift to automatically copy snapshots (automated or manual) for a cluster to another region. When a snapshot is created in the cluster's primary region, it will be copied to a secondary region known as the source region and destination region. By storing a copy of your snapshots in another region, you have the ability to restore your cluster from recent data if anything affects the primary.

Snapshots are point-in-time backups of a cluster. **There are two types of snapshots:** automated and manual. Amazon Redshift stores these snapshots internally in Amazon S3 using an encrypted Secure Sockets Layer (SSL) connection.

Amazon Redshift automatically takes incremental snapshots that track changes to the cluster since the previous automated snapshot. Automated snapshots retain all of the data required to restore a cluster from a snapshot. You can create a snapshot schedule to control when automated snapshots are taken, or you can take a manual snapshot at any time.

When you restore from a snapshot, Amazon Redshift creates a new cluster and makes the new cluster available before all of the data is loaded, so you can begin querying the new cluster immediately. The cluster streams data on demand from the snapshot in response to active queries then loads the remaining data in the background.